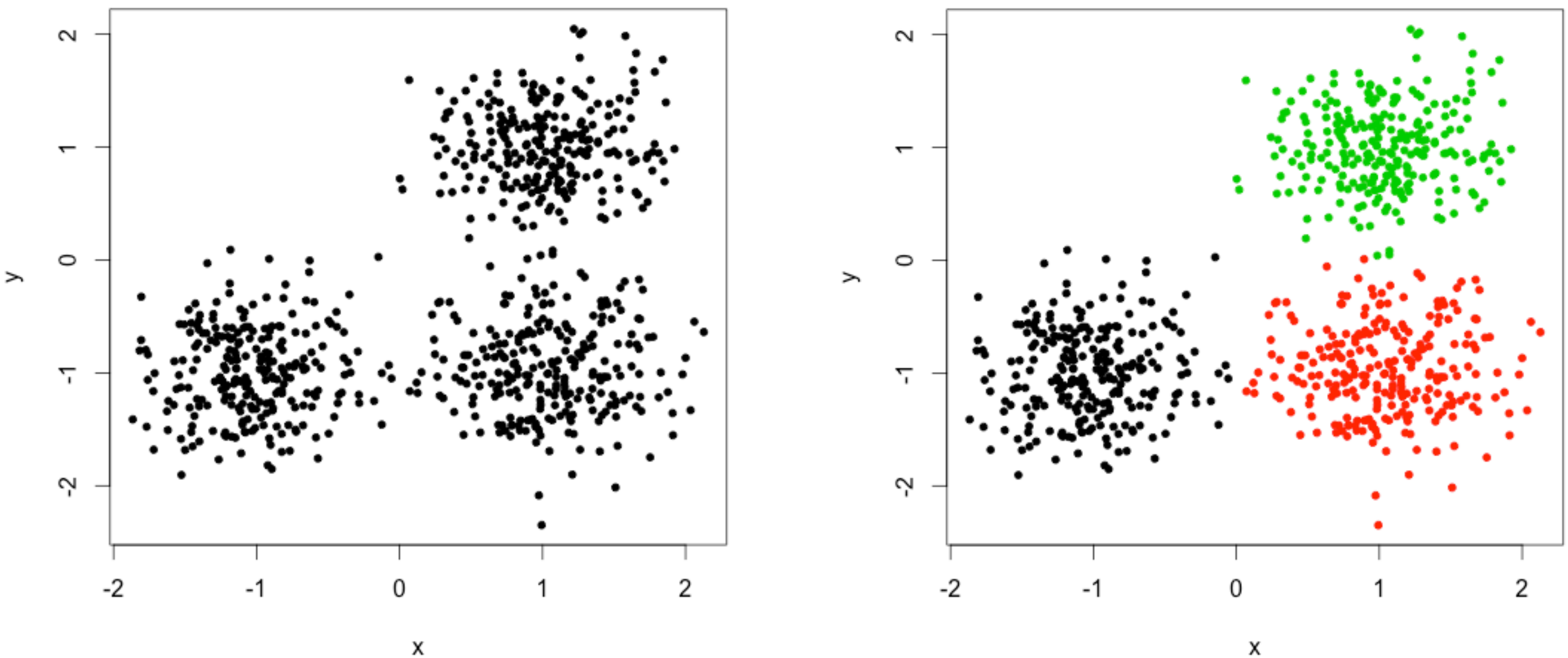
ARCHER Training Courses

Unsupervised Learning

Reusing this material

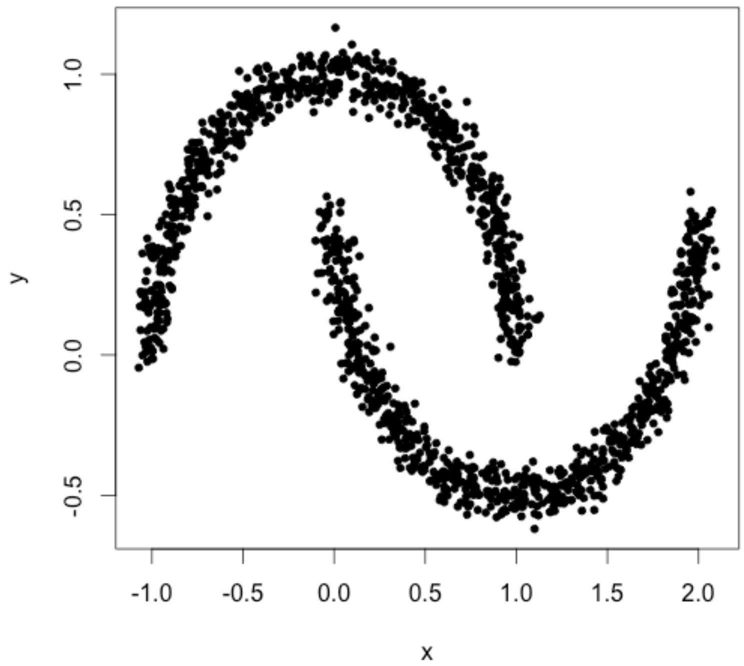
Clustering – the problem

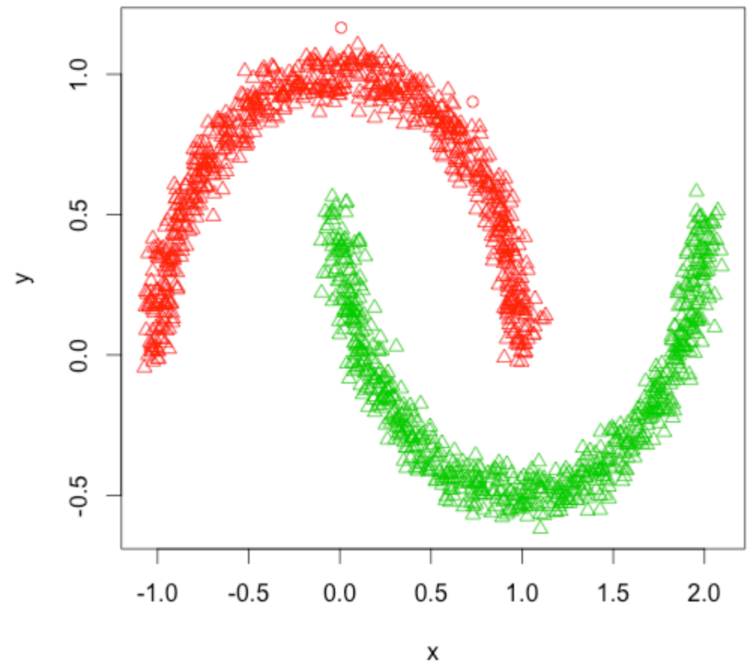
- Partitioning set of data points into groups which are "similar"
- Unsupervised learning



Clustering – the problem

- Partitioning set of data points into groups which are “similar”





archer

epcc

THE UNIVERSITY OF EDINBURGH

Clustering

- Also called segmentation, stratification, grouping
- Offer different experiences to different people in marketing
  - Young Urban Professionals, Double Income No Kids etc.
- Different models for different groups
- Different algorithms
  - k-means
  - Distribution based
  - Density based

K-means clustering

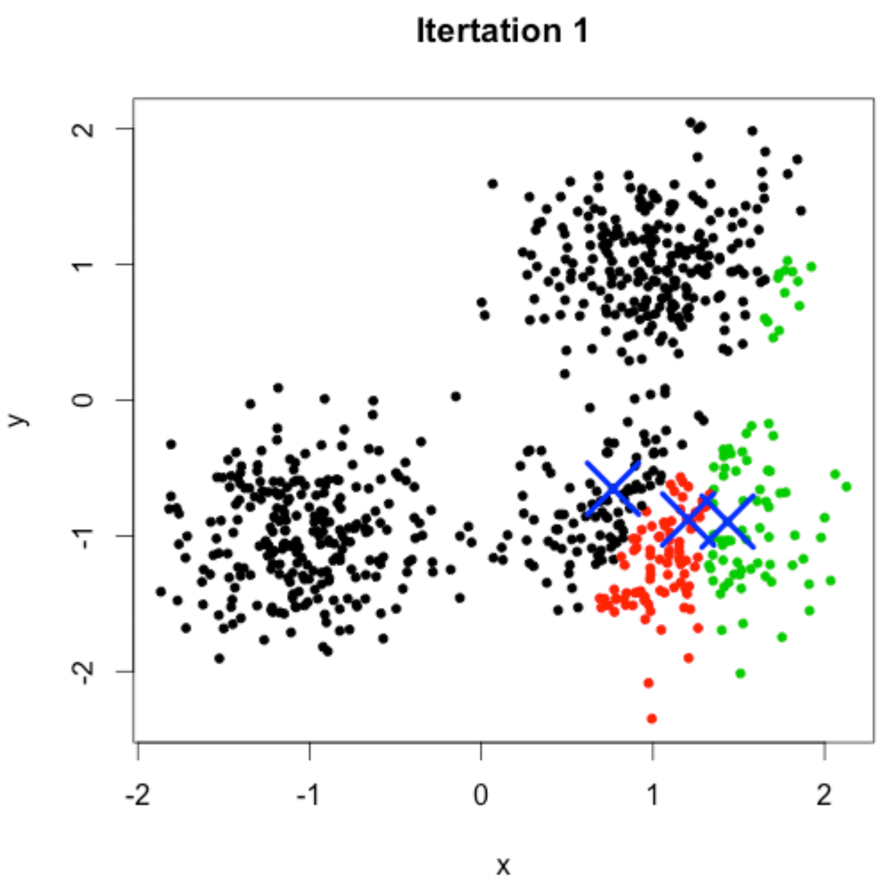
K-means clustering

- Know in advance that there are *k* clusters
- Goal:
  - Given observation vectors: $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m$
  - Group them in *k* distinct sets: $S_1, S_2, \ldots, S_k$
  - Minimise the within-cluster sum of squares

$$\sum_{i=1}^{k} \sum_{x \in S_i} \|\mathbf{x} - \mu_i\|^2$$

where μ_i is mean of points in S_i

K-means in action

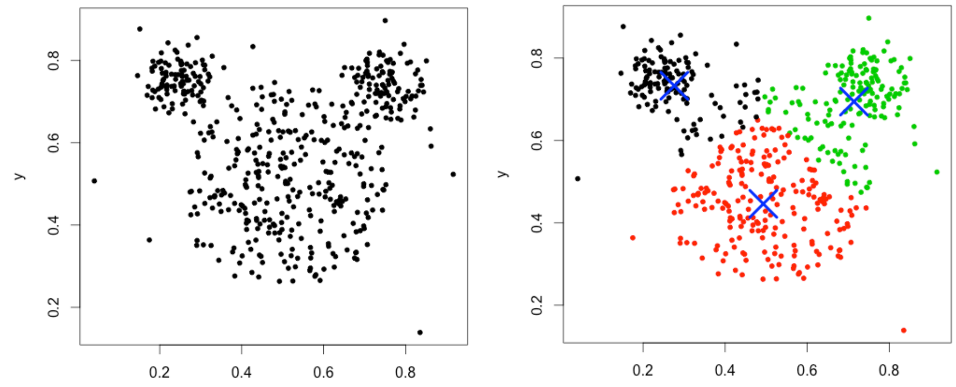


K-means in practice

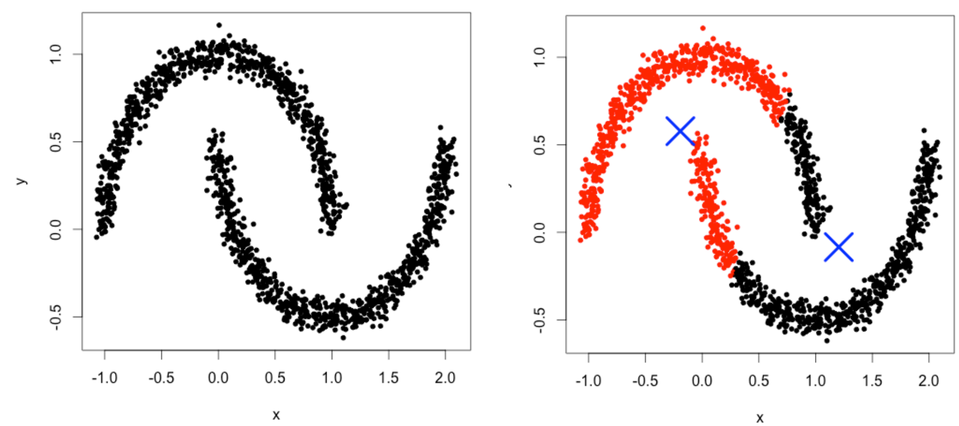
- Must normalise the features so that the distances are not biased to a particular dimension
- Need to think carefully about the features you wish to include as algorithm will give each feature equal weight
  - Unlike linear regression where the weight may be zero, or Naïve Bayes where a meaningless feature will have no real impact
- Can be hard to interpret
  - Sometimes the clusters seem meaningless
- Need to choose *k*
  - Sometimes you know there are *k* processes generating the data
  - Trial and error
  - Look for ‘knee’ in plot of cost against *k*

Limitation of k-means

- Clusters assumed to be the same size



- Clusters on density not so good

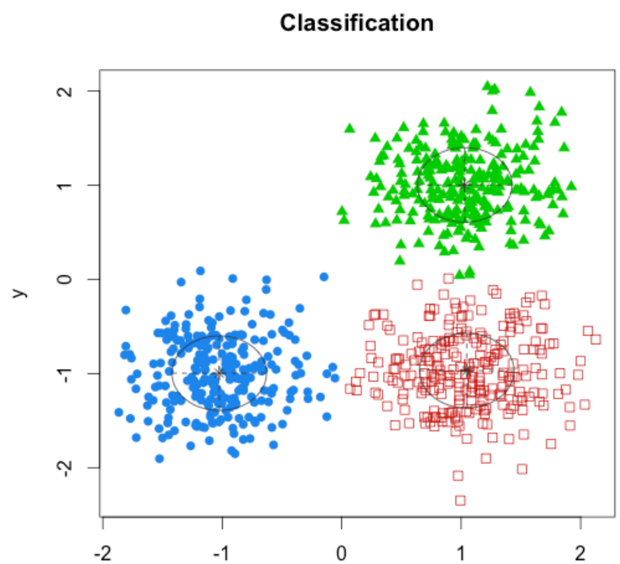


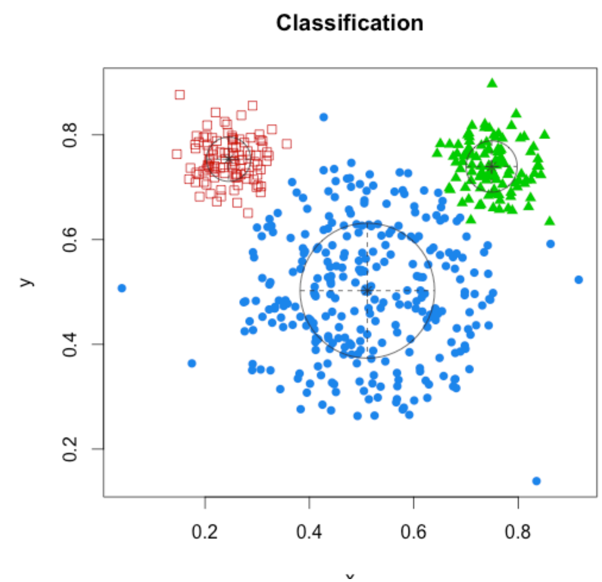
Distribution based clustering

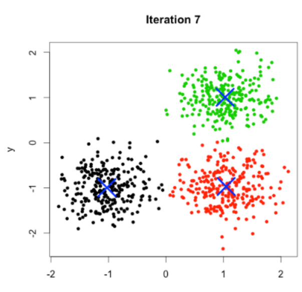
Distribution based clustering

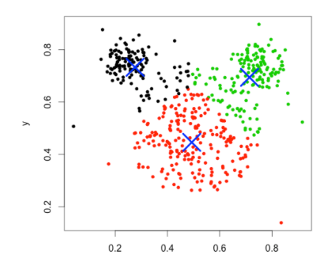
- Model the data using statistical distributions
- Gaussian mixture models
  - Model is a fixed number of Gaussian distributions
  - Need to discover the parameters of these Gaussian distributions
    - For each cluster need to know mean for each feature dimension and the covariance matrix
- Expectation maximization algorithm
  - Starts with random parameters and iteratively updates, scanning the whole data set on each iteration
  - Similar to *k*-means but maths beyond scope of this course
  - Finds a local optimum
- Data points are assigned to the distribution they most likely belong to (hard clusters), or each data point is given probability of belonging to clusters (soft clusters)

Distribution based clustering in practice



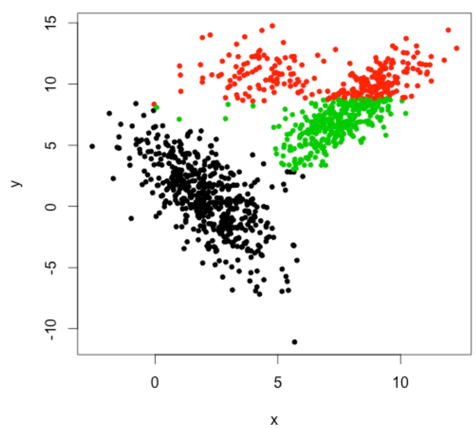


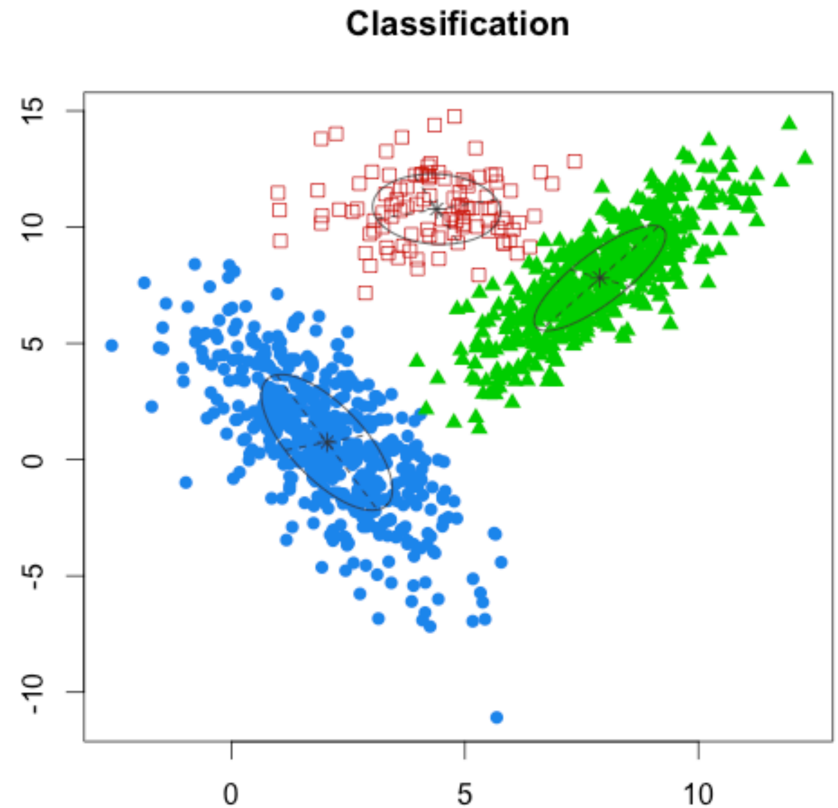




Distribution based clustering

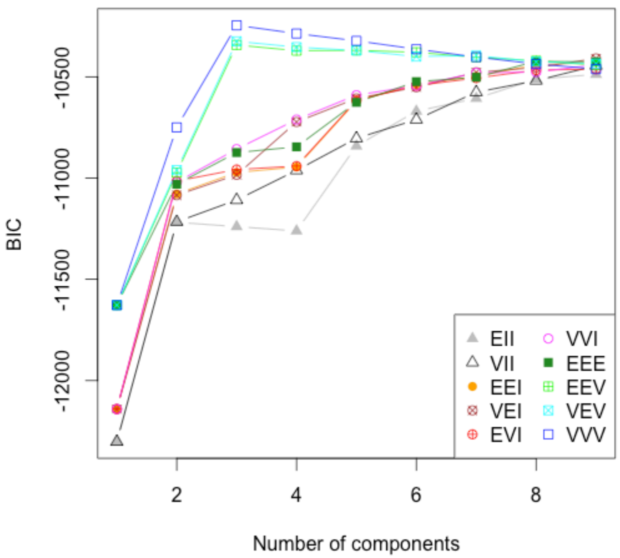
- Handles covariance of features
  - No need to normalise data

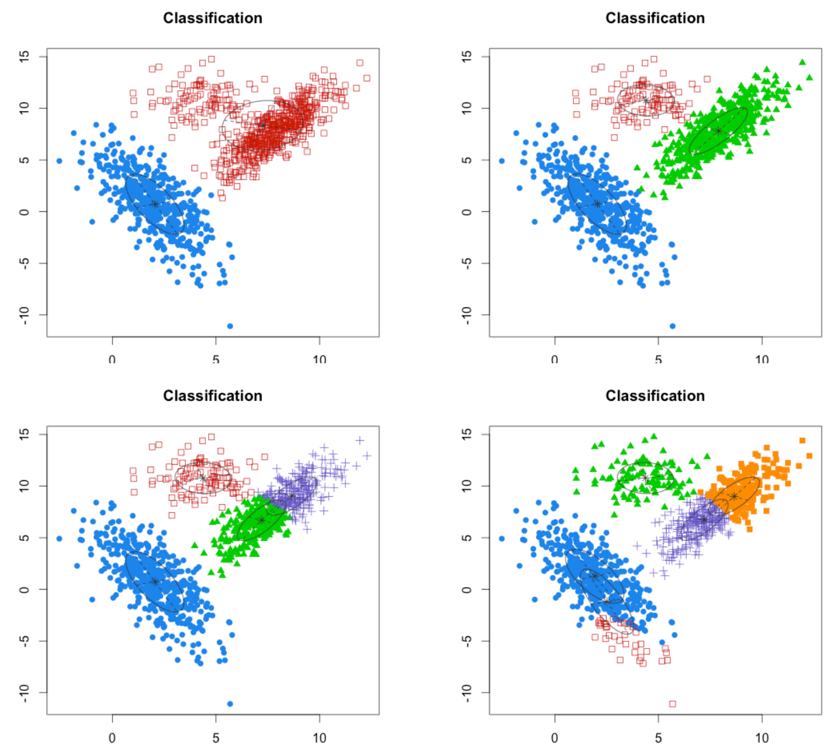




Distribution based clustering

- Can choose number of clusters





Model of covariance matrix.
(Note: k refers to feature dimensions rather than number of clusters)

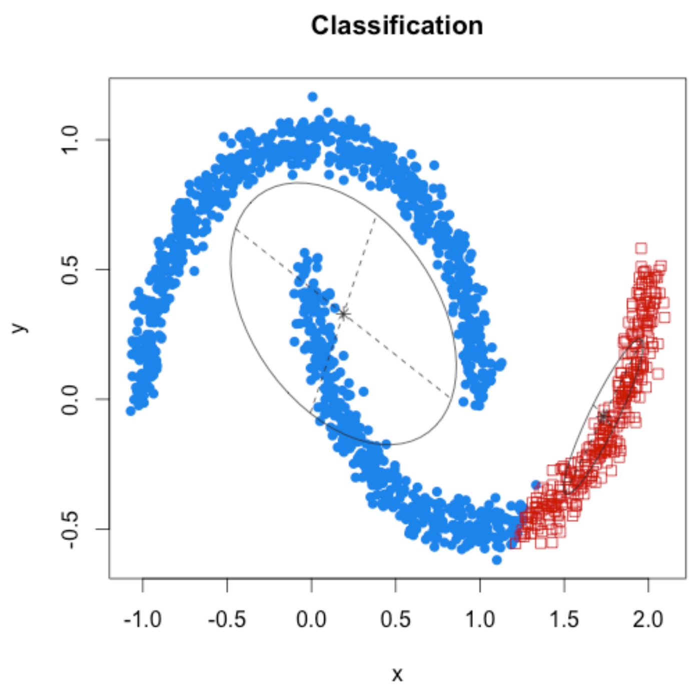
identifier	Model	HC	EM	Distribution	Volume	Shape	Orientation
E		•	•	(univariate)	equal		
V		•	•	(univariate)	variable		
EII	λI	•	•	Spherical	equal	equal	NA
VII	$\lambda_k I$	•	•	Spherical	variable	equal	NA
EEI	λA		•	Diagonal	equal	equal	coordinate axes
VEI	$\lambda_k A$		•	Diagonal	variable	equal	coordinate axes
EVI	λA_k		•	Diagonal	equal	variable	coordinate axes
VVI	$\lambda_k A_k$		•	Diagonal	variable	variable	coordinate axes
EEE	$\lambda D A D^T$	•	•	Ellipsoidal	equal	equal	equal
EEV	$\lambda D_k A D_k^T$		•	Ellipsoidal	equal	equal	variable
VEV	$\lambda_k D_k A D_k^T$		•	Ellipsoidal	variable	equal	variable
VVV	$\lambda_k D_k A_k D_k^T$	•	•	Ellipsoidal	variable	variable	variable

BIC = Bayesian Information Criterion

- Score based on fit of model to data with increasing penalty for more clusters

Limitations of distribution based clustering

- Bad for density based clusters that don’t match distribution model



Density based clustering

Density based clustering

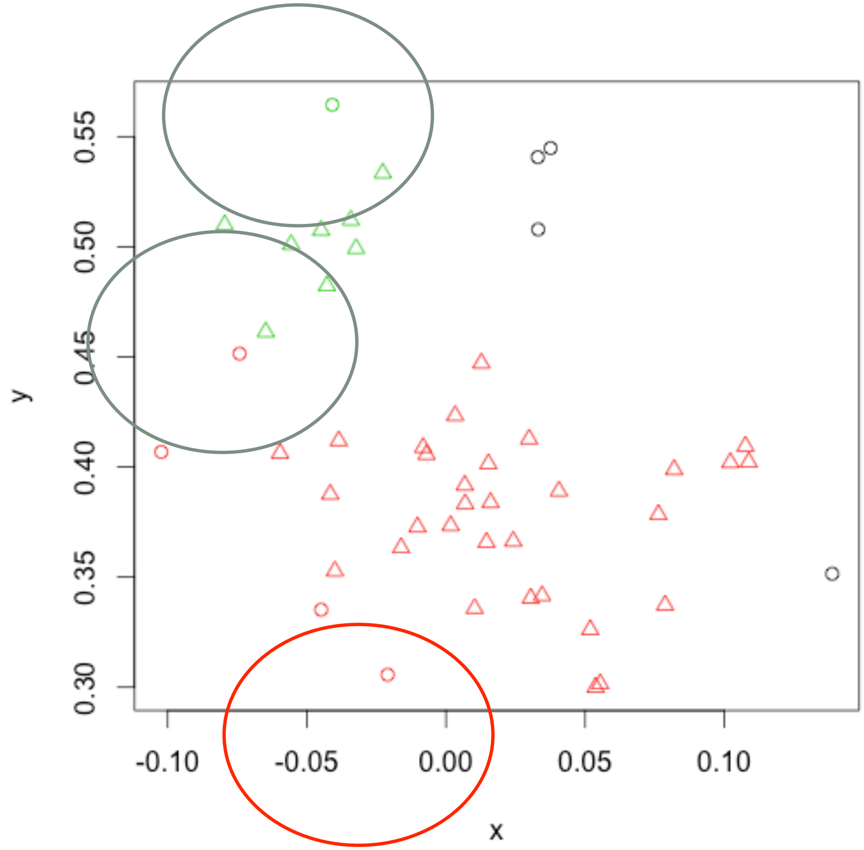
- Clusters are defined as areas of higher density than the rest of the data set
  - Points in sparse areas are considered noise and border points

- Popular method is DBSCAN algorithm:
  - Density-Based Spatial Clustering of Applications with Noise
  - Group together points with many nearby neighbours
  - Points with few nearby neighbours are marked as outliers
  - Two parameters:
    - ε : distance below which points are considered neighbours
    - *minPts:* minimum number of points required to form a cluster
  - Uses “density-reachability” cluster model

DBSCAN definition

- All points are identified as one of:
  - Core point:
    - A point *p* with at least *minPts* points within *ε* of it
    - Those points within *ε* of *p* are *directly-reachable* from *p*
  - Density-reachable point:
    - A point *q* is reachable from *p* if there is a path $p_1, \ldots, p_n$ where $p_1 = p,\ p_n = q$ and p_{i+1} is directly reachable from p_i
  - Outlier
    - Point not reachable from any other point
- Points *p* and *q* are density connected if there exists a point *o* such that *p* and *q* are density-reachable from *o*.
- A cluster defined as:
  - Containing all points that are mutually density-connected
  - Also contains any points density-reachable from a point in cluster

DBSCAN



DBSCAN: $minPts = 5, \varepsilon = 0.05$

DBSCAN algorithm

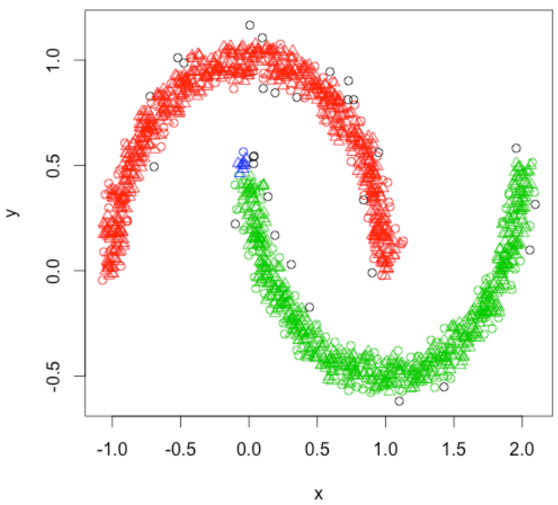
```
DBSCAN(D, eps, MinPts)
   C = 0
   for each unvisited point P in dataset D
      mark P as visited
      NeighborPts = regionQuery(P, eps)
      if sizeof(NeighborPts) < MinPts
         mark P as NOISE
      else
         C = next cluster
         expandCluster(P, NeighborPts, C, eps, MinPts)

expandCluster(P, NeighborPts, C, eps, MinPts)
   add P to cluster C
   for each point P' in NeighborPts
      if P' is not visited
         mark P' as visited
         NeighborPts' = regionQuery(P', eps)
         if sizeof(NeighborPts') >= MinPts
            NeighborPts = NeighborPts joined with NeighborPts'
      if P' is not yet member of any cluster
         add P' to cluster C

regionQuery(P, eps)
   return all points within P's eps-neighborhood (including P)
```

Exactly one call to regionQuery for each point.

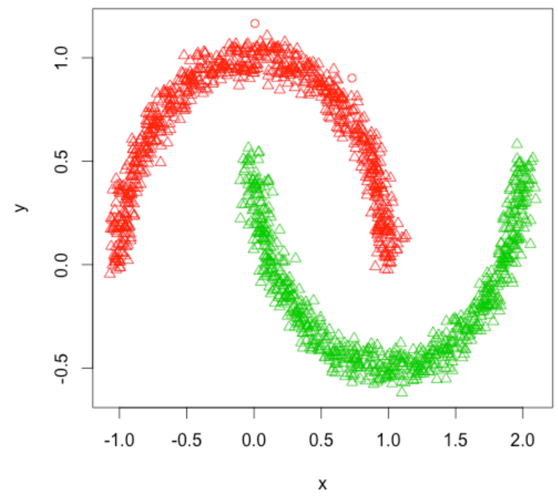
If indexed this call is O(log*n*) so whole algorithm is O(*n*log*n*)

Algorithm from wikipedia: http://en.wikipedia.org/wiki/DBSCAN

DBSCAN handles density clusters

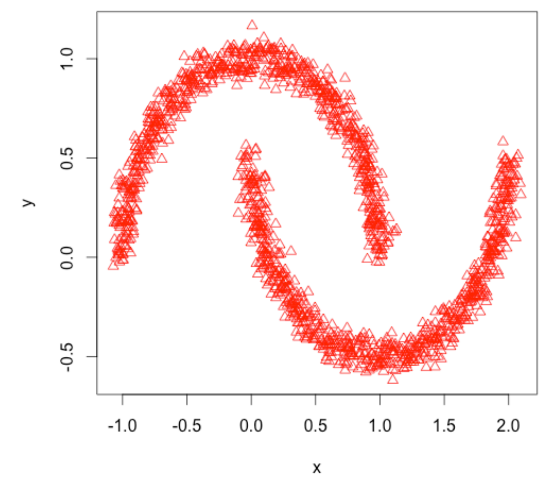


$minPts = 5,\ \varepsilon = 0.05$



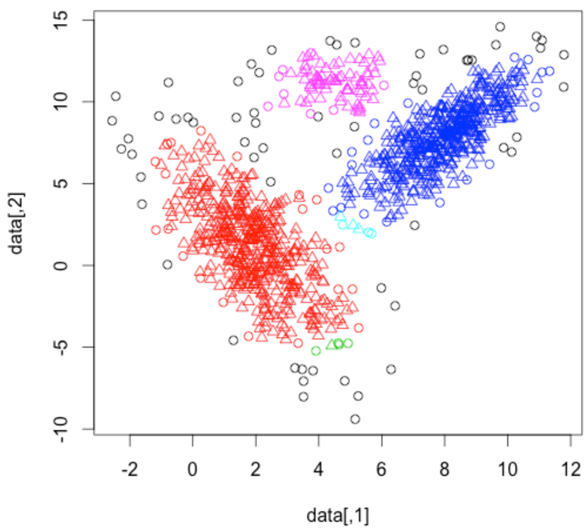
$minPts = 5, \varepsilon = 0.1$

$minPts = 5, \varepsilon = 0.2$

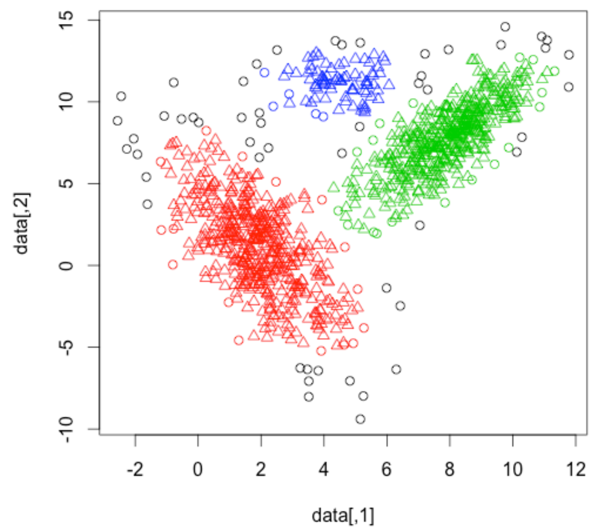


$minPts = 5, \varepsilon = 0.4$

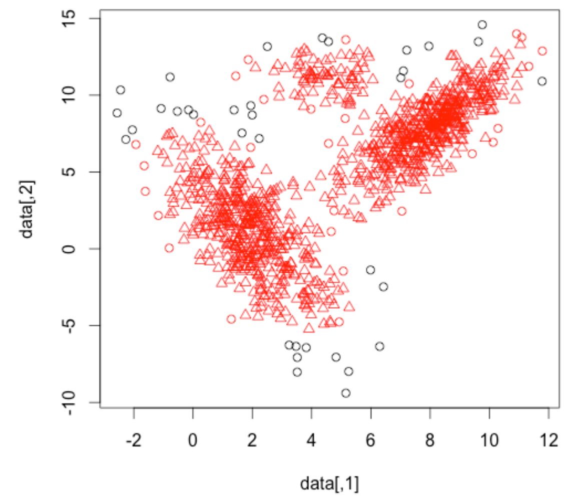
DBSCAN limitations



$minPts = 5, \varepsilon = 0.7$



$minPts = 5, \varepsilon = 0.8$



$minPts = 5, \varepsilon = 0.9$

- Can be hard to tune parameters
- Does not produce model that can be used to classify data
- Cannot handle clusters with highly variable densities

HPC implementations

HPC implementations

- *k*-means supports a data-parallel implementation:
  - All nodes get subset of the data
  - Repeat
    - Centroids sent to all nodes
    - Points assigned to nearest centroid
    - For each cluster feature sums and count returned to master
    - Master computes new centroids
  - Until centroids are stable

- SPRINT implementation of Partitioning Around Medoids (PAM)

Streaming implementation of k-means

- Single pass through data – low memory overhead:
- Keep *k* weighted centroids:
  - While more data
    - If new point ‘*close*’ to existing centroid then add to that cluster, else create new cluster
    - When number of clusters beyond *k*
      - Increase definition of ‘*close*’
      - Re-centre the clusters
      - Stochastically merge clusters together, preferring to merge smaller clusters that are close together.
  - Run a *k*-means on the weighted centroids

Streaming k-means in action

- Movie does not include the final *k*-means stage
- Frames of movie only at points where clusters change

Streaming k-means continued

- How do we record which final cluster a data point belongs to?
  - Do a second pass with the final *k* centroids
  - Give each cluster a unique identifier and record cluster mergers
    - DataPoint 1 -> cluster 1
    - DataPoint 3452 -> cluster 5
    - Final cluster f1: 1, 4, 24, 72, 8, 78, 67 29
    - Final cluster f2: 7, 38, 27, 442, 338

Advanced clustering

Advanced clustering

- **Probabilistic Topic modelling**
  - Topics are groups of related words with a probability for each word
    - (gene 0.04, dna 0.02, …) (data 0.02, computer 0.01, ...)
  - Documents are made up from a collection of topics with different probabilities
    - (“genetics” 0.3, “computers”, 0.2, “government”, 0.01, …)
  - Words within document come from the topics at the specified probability and then from within the topic at the specified probability
  - Can then use algorithms such as Latent Dirichlet Allocation to extract the topics for a collection of documents

Probabilistic topics modelling - Example

- Associated Press data from the First Text Retrieval Conference (TREC-1) 1992.

```
> Terms<-terms(lda, 10) #10 first terms of each topic ordered by frequency
> Terms
      Topic 1    Topic 2    Topic 3  Topic 4      Topic 5     Topic 6   Topic 7     Topic 8          Topic 9    Topic 10
 [1,] "school"   "trade"    "i"      "soviet"     "police"    "percent" "company"   "bush"           "court"    "program"
 [2,] "new"      "late"     "just"   "government" "people"    "year"    "million"   "president"      "case"     "people"
 [3,] "years"    "oil"      "dont"   "united"     "two"       "million" "workers"   "house"          "attorney" "report"
 [4,] "first"    "states"   "like"   "president"  "killed"    "billion" "new"       "dukakis"        "law"      "state"
 [5,] "students" "united"   "people" "party"      "miles"     "market"  "corp"      "campaign"       "judge"    "health"
 [6,] "wife"     "dollar"   "time"   "minister"   "three"     "last"    "billion"   "committee"      "office"   "children"
 [7,] "family"   "new"      "think"  "union"      "officials" "stock"   "business"  "administration" "federal"  "years"
 [8,] "show"     "cents"    "going"  "states"     "spokesman" "prices"  "inc"       "congress"       "state"    "national"
 [9,] "black"    "iraq"     "get"    "official"   "city"      "sales"   "pay"       "bill"           "charges"  "system"
[10,] "world"    "thursday" "day"    "political"  "reported"  "new"     "employees" "reagan"         "trial"    "public"
```

Machine Learning wrap-up

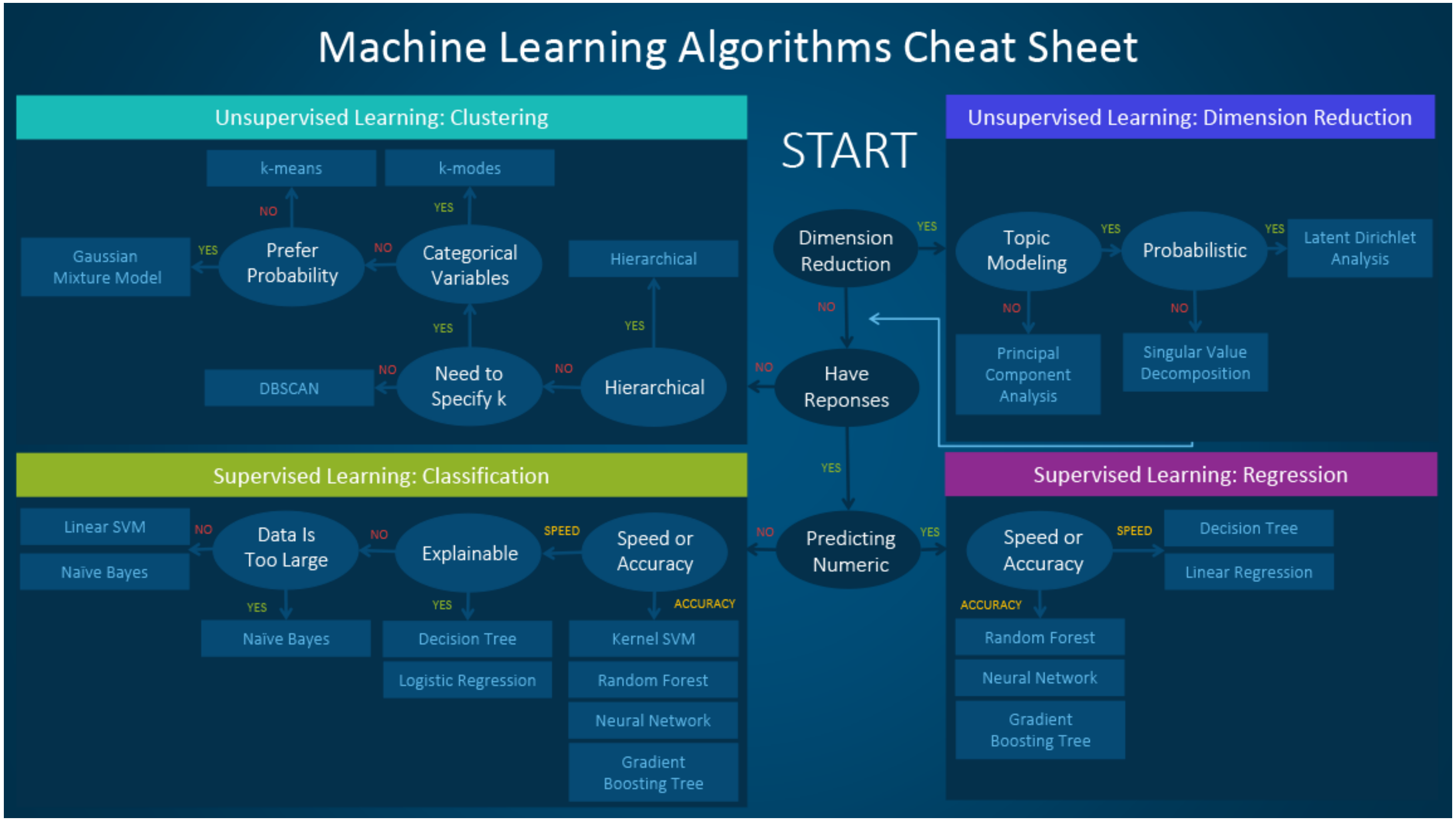


Image source: http://www.kdnuggets.com/2017/06/which-machine-learning-algorithm.html

archer

epcc

THE UNIVERSITY OF EDINBURGH